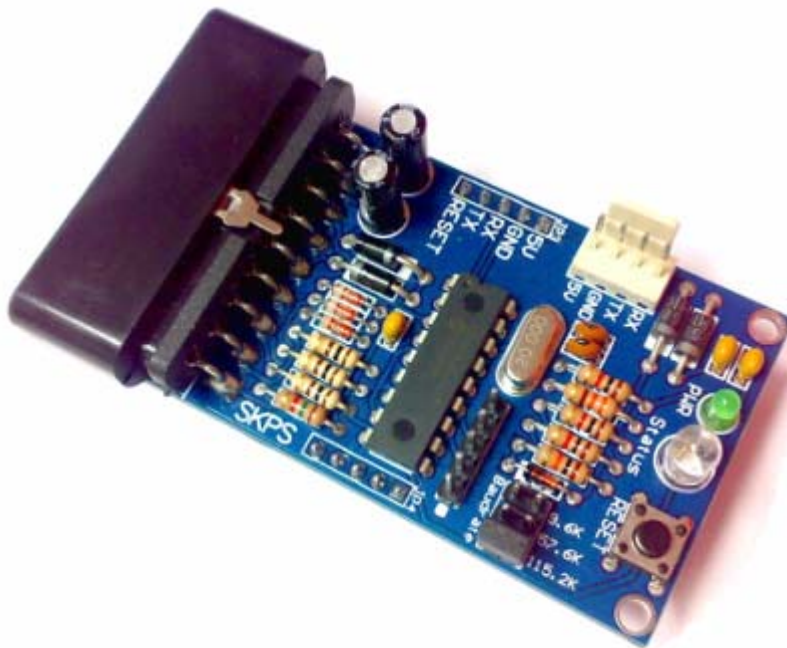




PS2 Controller Starter Kit

SKPS



User's Manual

V1.0

Oct 2008

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Index

1. Introduction	1
2. System Overview	2
3. Packing List	3
4. Board Layout	4
5. Product Specification and Limitations	5
6. Hardware Interface	6
6.1 Microcontroller	6
6.2 Computer	8
6.3 PS2 Controller	14
7. Protocol	19
7.1 Non-ASCII Mode (Microcontroller)	19
7.1.1 Button and PS2 Controller Status	19
7.1.2 On Board Vibrator Motor Control	20
7.2 ASCII Mode (Computer)	21
7.2.1 Button and PS2 Controller Status	21
7.2.2 Read Key Functions	22
7.2.3 On Board Vibrator Motor Control	23
8. Getting Started	24
9. Warranty	27

1. INTRODUCTION

Play Station 2 (PS2) controller is relatively easy to obtain from any game store and it offers good human manual input for control system. More and more developers are looking into applying existing PS2 controller to control particular system. The major problem to archive this is the socket for PS2 and the protocol to communicate with it. PS2 socket is very unique and difficult to source. Besides, protocol to obtain the status (digital and analog) of each button and analog stick on PS2 controller create headache to developer.

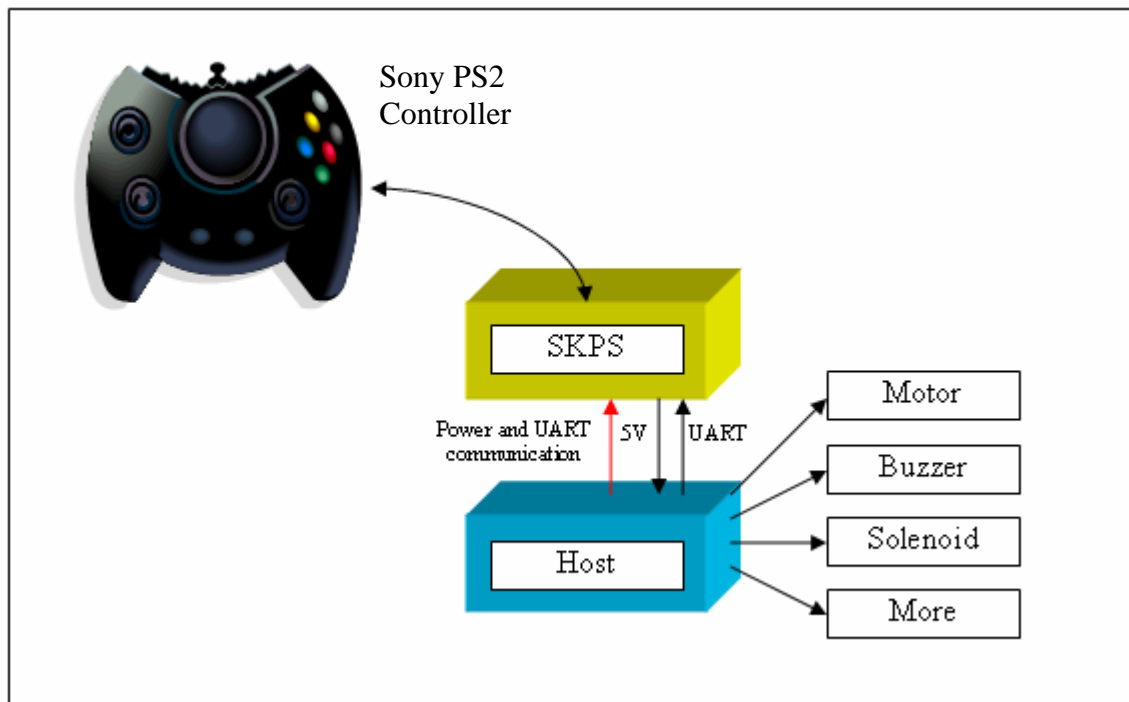
Thus, Cytron Technologies has designed a PS2 Controller Starter Kit, SKPS to offer solution. This starter kit offers a compact yet reliable PS2 Controller Converter for user. SKPS's power and UART pin is compatible with SKKCA and SKXBee. A host is needed to communicate with SKPS through UART. Reading Joy-stick button's state will be as easy as 1, 2, 3. It offers a standard connector for Sony PS2 controller to plug-in.

It has been designed with capabilities and features of:

- 5V powered, low current consumption, less than 150mA.
- 1 standard PS2 controller socket.
- Communicate with host microcontroller through **UART**.
- Simple inquiry command and button status feedback for host to process.
- **Vibrator motor** is controllable.
- **Wired or wireless PS2** controller is fully compatible.
- PS2 Controller will only operate in analog mode.
- A small LED as indicator for communication between SKPS and PS2 controller.
- **Dimension** 7.4cm x 3.9cm

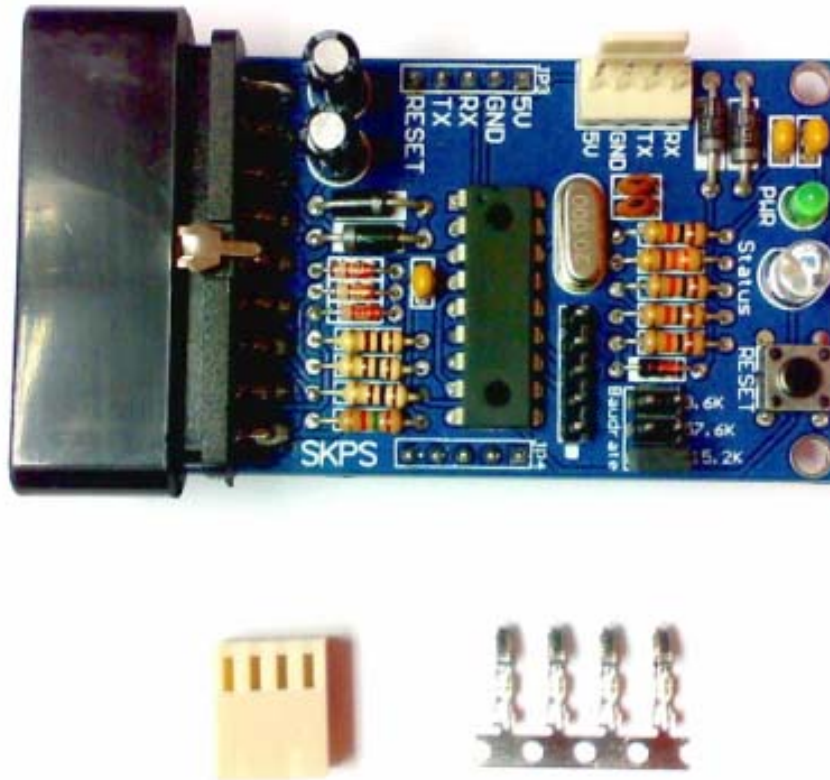
NOTE: Communication between SKPS and host is done through 5V TTL UART at selectable baud rate (9600, 57600 or 115200)

2. SYSTEM OVERVIEW



3. PACKING LIST

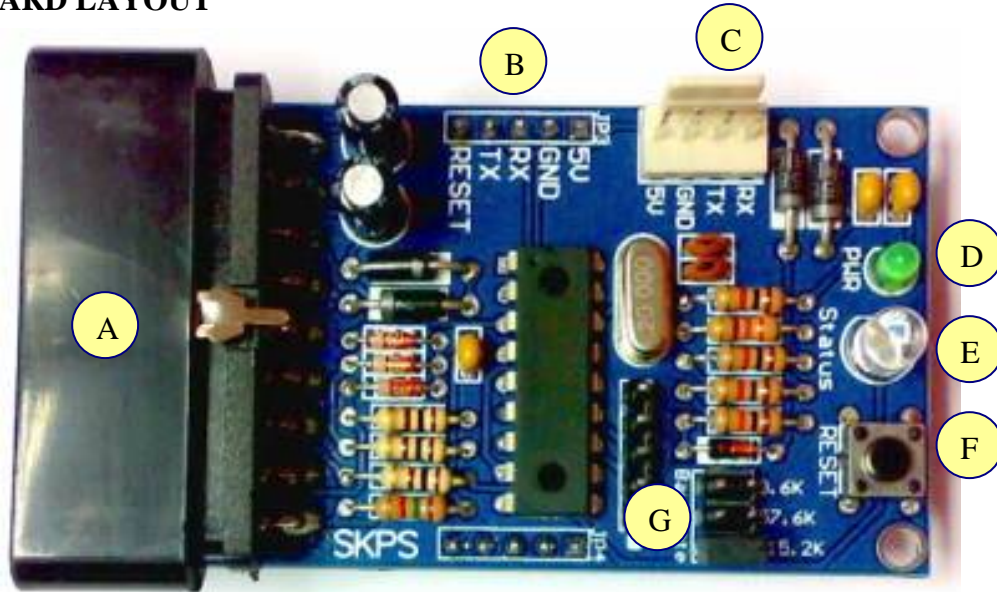
Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.



1. 1 x SKPS board.
2. 1 x mini jumper of SKPS board.
3. 4 x 2510 iron pin.
4. 1 x 2510 4 ways female connector.

Note: SKPS does not come with PS2 controller, please purchase separately from Cytron Technologies website. It is advised to use PS2 controller from Cytron Technologies because all PS2 controller provided is tested before it is being shipped to customer.

4. BOARD LAYOUT



Label	Function
A	PS2 connector socket, please connect the PS2 controller here
B	5 ways header pin for external power and UART interface to microcontroller.
C	4 ways 2510 header pin for external power and interface to PC or microcontroller also (depends to user).
D	On board power indicator LED (Green)
E	On board indicator LED (Blue) indicators for check connection with controller status.
F	On board reset button for SKPS.
G	Baud rate selector.

A – Connector for PS2 controller

B – 5 ways header pin for external power supply and UART interface.

C – Alternative 4 ways 2510 header pin for external power supply and UART.

D – On board power indicator LED. It should be ON if SKPS is powered by stable 5V.

E – On board indicator LED, it indicators the connection status between SKPS and PS2 controller.

No controller connected	: Blink
Controller connected	: On with 50% brightness
Controller connected (button pressed):	On with 100% brightness

F – Reset button for SKPS board.

G – Baud rate selector. User may either choose 9600, 57600 or 115200 baud rate. Place the jumper to select the baud rate.

5. PRODUCT SPECIFICATION AND LIMITATIONS

SKPS is designed to offer a compact and reliable PS2 Controller Converter for user. The specifications are as listed below:

Label	Definition	Function
5V	Power Input for SKPS	External power source for SKPS, the typical voltage is 5V. Please ensure this 5V source is a stable supply. Please do not use normal AC-DC adaptor to power it. It is recommended to use linear regulator (7805) to provide 5V supply.
GND	Ground or negative	Ground of power and signal.
RX	SKPS UART Receive signal	This is SKPS's receiver pin, it should be interfaced to 5V logic UART, no divider is necessary. This is an input pin to SKPS. It should be connected to microcontroller's transmitter pin.
TX	SKPS UART Transmit signal	This is SKPS's transmitter pin; it should be interfaced to 5V logic UART. This is an output pin from SKPS. It should be connected to microcontroller's receiver pin.
RESET	SKPS Reset pin	Reset pin of SKPS. It should be connected to a push button to Gnd, or NPN transistor.

Absolute Maximum Rating

Symbol	Parameter	Min	Max	Unit
5V	Power source for SKPS	4.5	5.5	V
GND	Operating voltage ground	0	0	V
RX	Receiver pin of SKPS	0	5.5	V
TX	Transmitter pin of SKPS	0	5.5	V
RESET	Reset pin of SKPS	0	5.5	V

NOTE: SKPS can only be powered by either by PC through 4 way 2510 connector or external power (5V). The 5V must be a stable supply. Any ripple higher than 5.5V will spoil the controller on SKPS and it is not replaceable.

6. HARDWARE INTERFACE

Generally, there are 2 methods of using SKPS. It has been designed for interface to embedded system with 5V TTL (microcontroller) or connection to computer (normally for functionality test).

6.1 Microcontroller

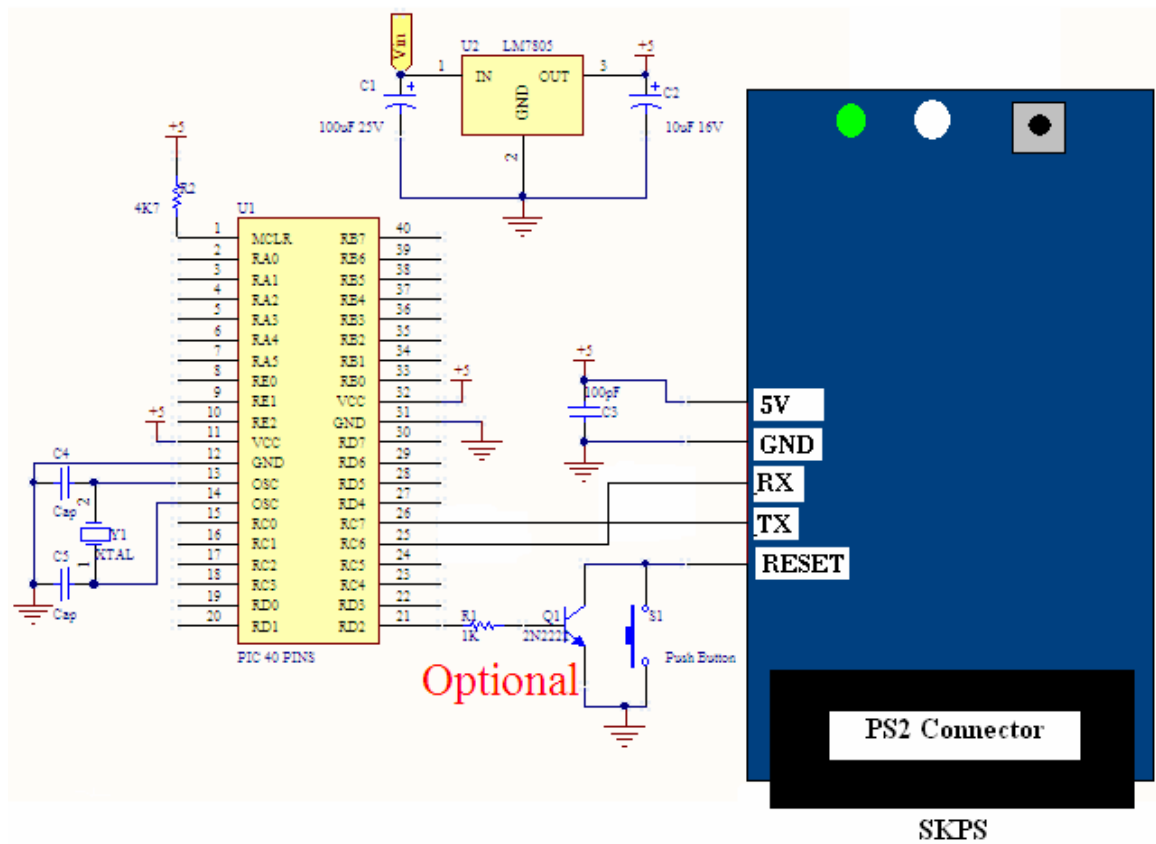
One of user's main concerns when using SKPS is the interface with microcontroller.

- a. To begin, user may connect 5V and Gnd of SKPS to microcontroller board. Header socket can be used to connect SKPS to microcontroller board.



- b. Once the 5V is supply to SKPS, the small green LED should light ON and Status LED will blink. After PS2 is connected, it will stop blink and ON with half brightness.
- c. As 2nd step, user might need to connect the RX and TX pin to microcontroller. Of course, these two pins should be cross connected to microcontroller. In other words, RX should be connected to microcontroller's Transmitter pin (TxD), while TX should be connected to microcontroller's Receiver pin (RxD). No extra component is necessary between these connections. For details connection, please refer to sample schematic.
- d. If configuration of UART is completed, SKPS is ready for embedded development. Sending and receiving data require software or firmware development on particular microcontroller.
- e. Finally, the RESET pin of SKPS. This is an optional pin for user as there is already a reset button on SKPS. However, if user would like the microcontroller to reset SKPS during run time, a transistor is required for interface between microcontroller and SKPS. Please refer to following schematic for example of microcontroller interfacing with SKPS.

- f. Any microcontroller with UART peripheral can be used to interface with SKPS.

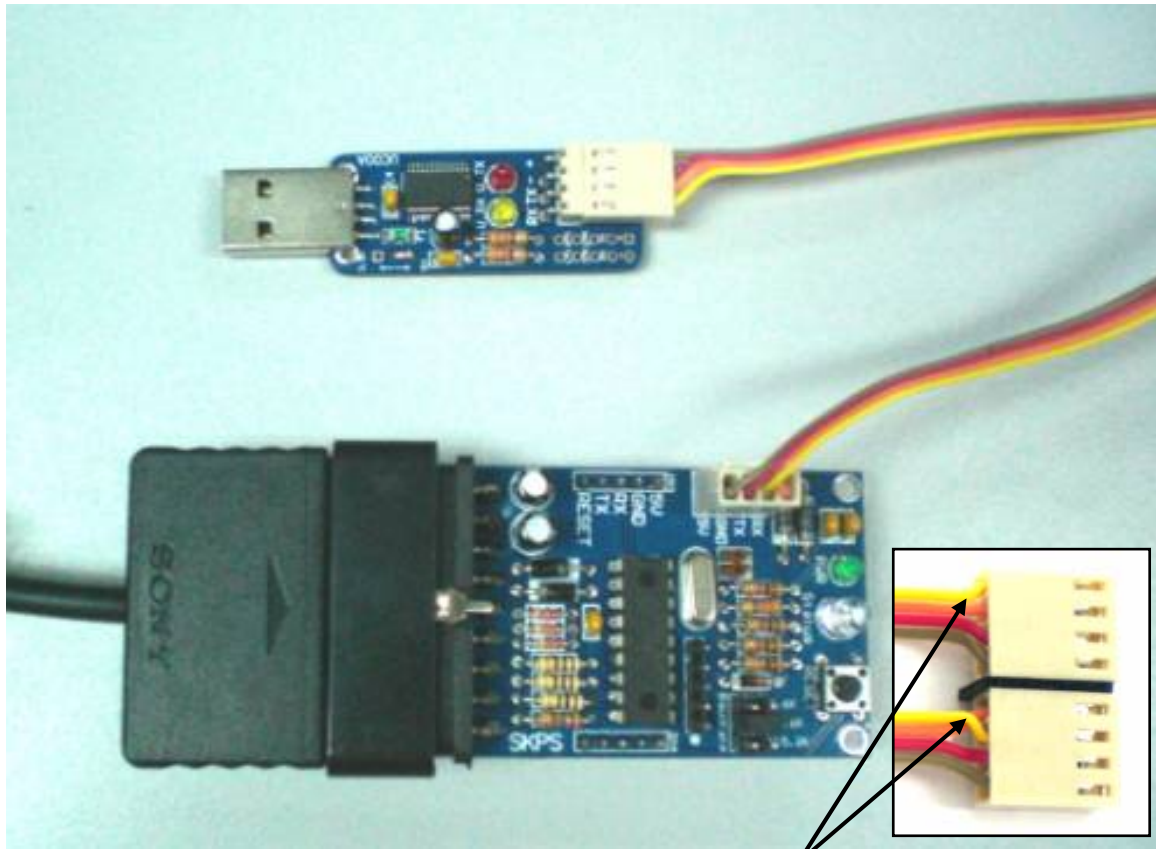


Example of connection to PIC16F877A microcontroller

6.2 Computer

SKPS can be connected to PC for functionality test. Another main concern during development using SKPS is to check the functionality. Normally, user will need to develop RS232 level shifter for communication to serial port. This generates extra work just to check the functionality of SKPS. Furthermore, laptop and computer nowadays have phase-out the serial port, USB have replaced it. With these reasons, an USB to UART converter have been developed. Now, no extra work is required to check the functionality of on board SKPS. Simply plug SKPS to UC00A and USB port of computer (PC or Laptop), install driver (1st time) and there is an extra virtual COM port ready for SKPS. Checking functionality is simple as 1, 2, 3.

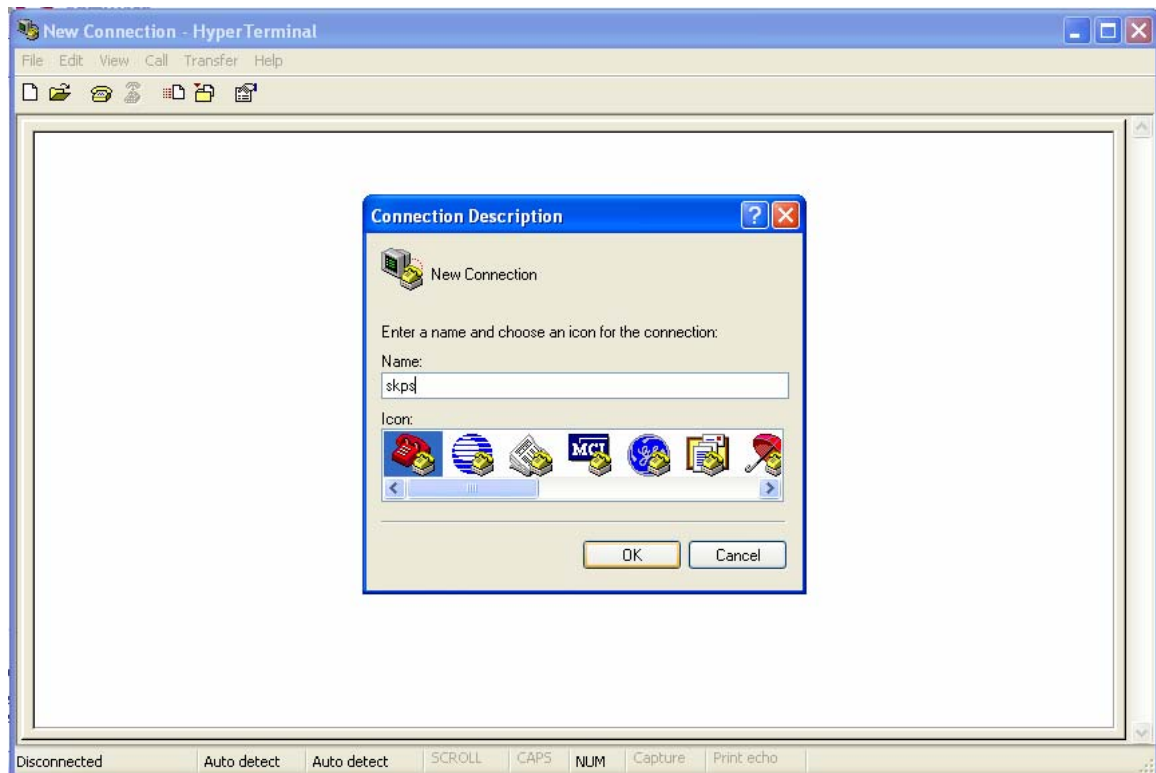
- a. Simply connect UC00A to SKPS, another end (A type) of UC00A to PC as shown in following figure. User might need to connect the RX and TX pin to UC00A. Of course, these two pins should be cross connected to UC00A. In other words, RX should be connected to UC00A's Transmitter pin (Tx), while TX should be connected to UC00A's Receiver pin (Rx). No extra component is necessary between these connections.



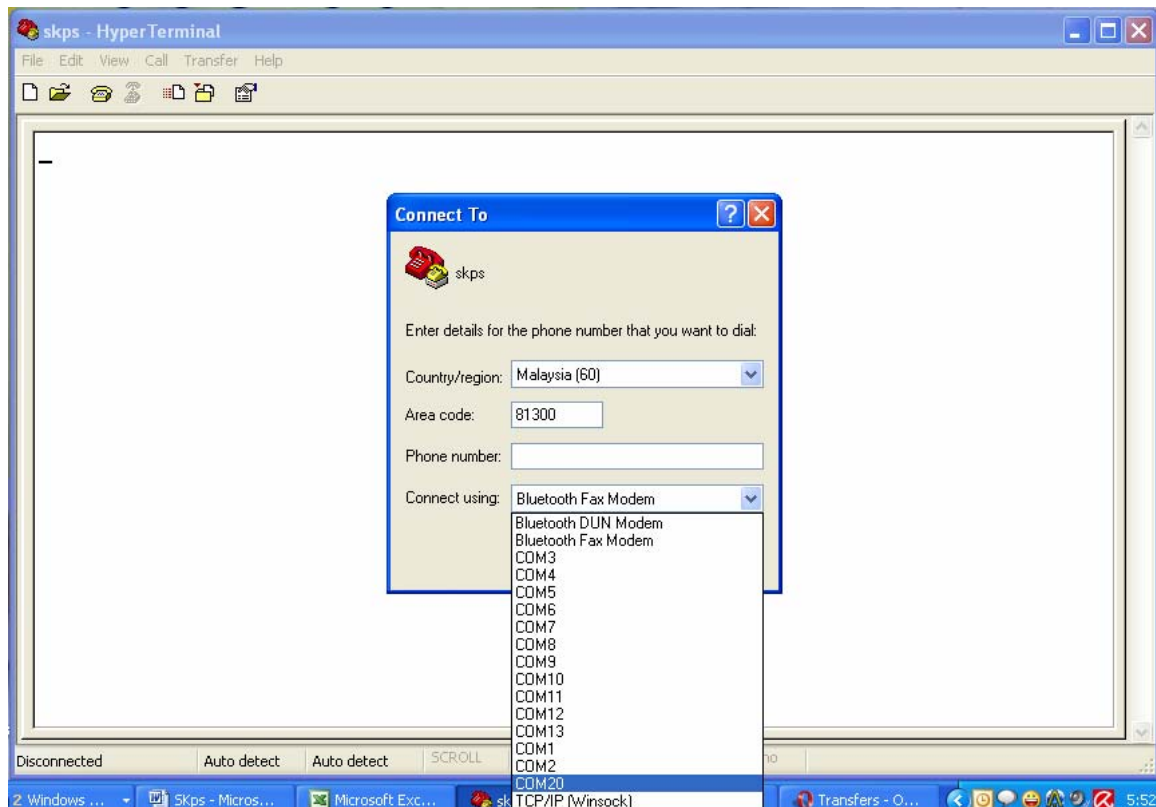
Wires in pin 1 and 2 are crossed for the second connector.

- b. Please refer to document named “USB Driver Installation Guide” for driver installation.

- c. After plug in the SKPS to computer and installation of driver, user is ready to test the functionality of SKPS. Open the HyperTerminal. Enter a name and choose an icon for connection as picture below then click OK.



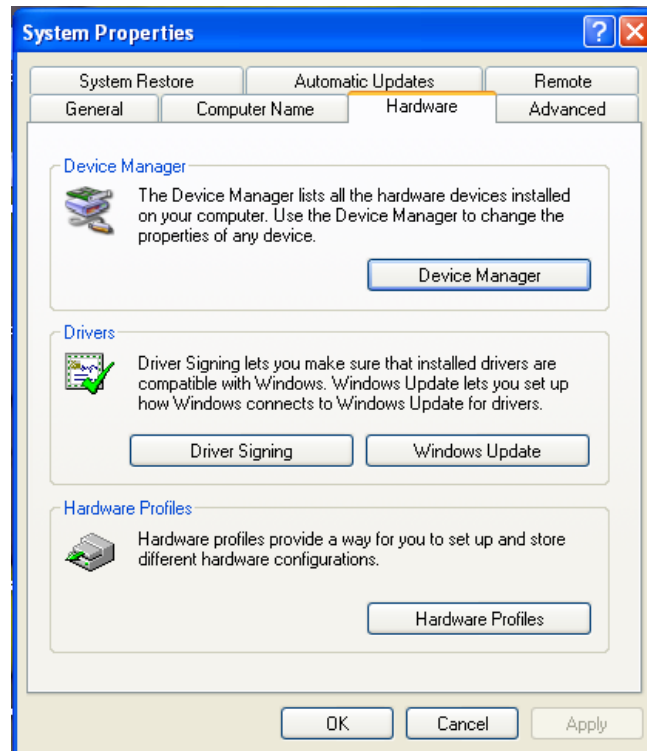
- d. Connect using USB Serial Port. If you are not sure which COM is it, please follow step (e) to (g).



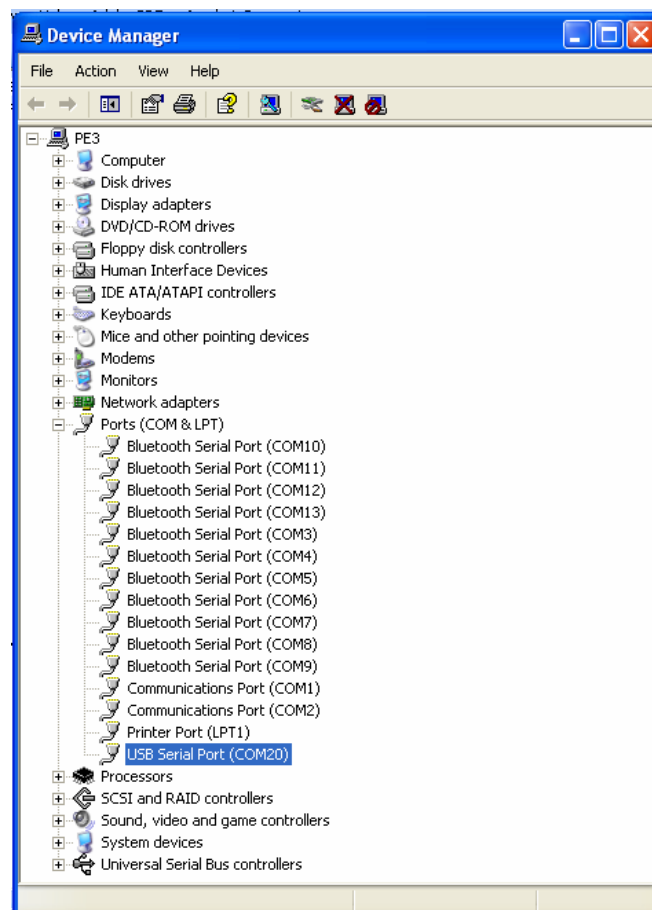
- e. Go to Start, right click on My Computer and choose Properties.



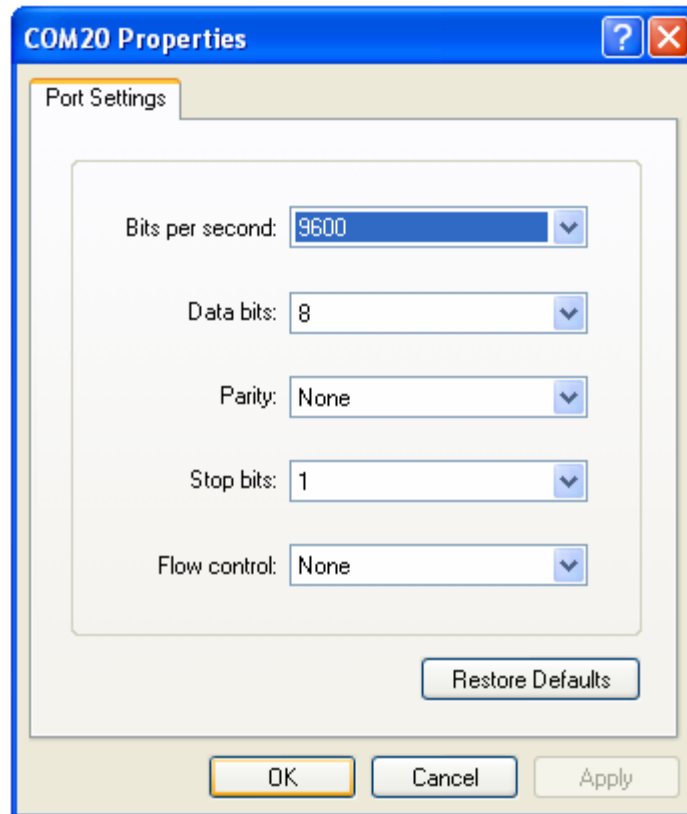
- f. At System Properties Table, choose Hardware and click on Device Manager. Device Manager Table will show out.



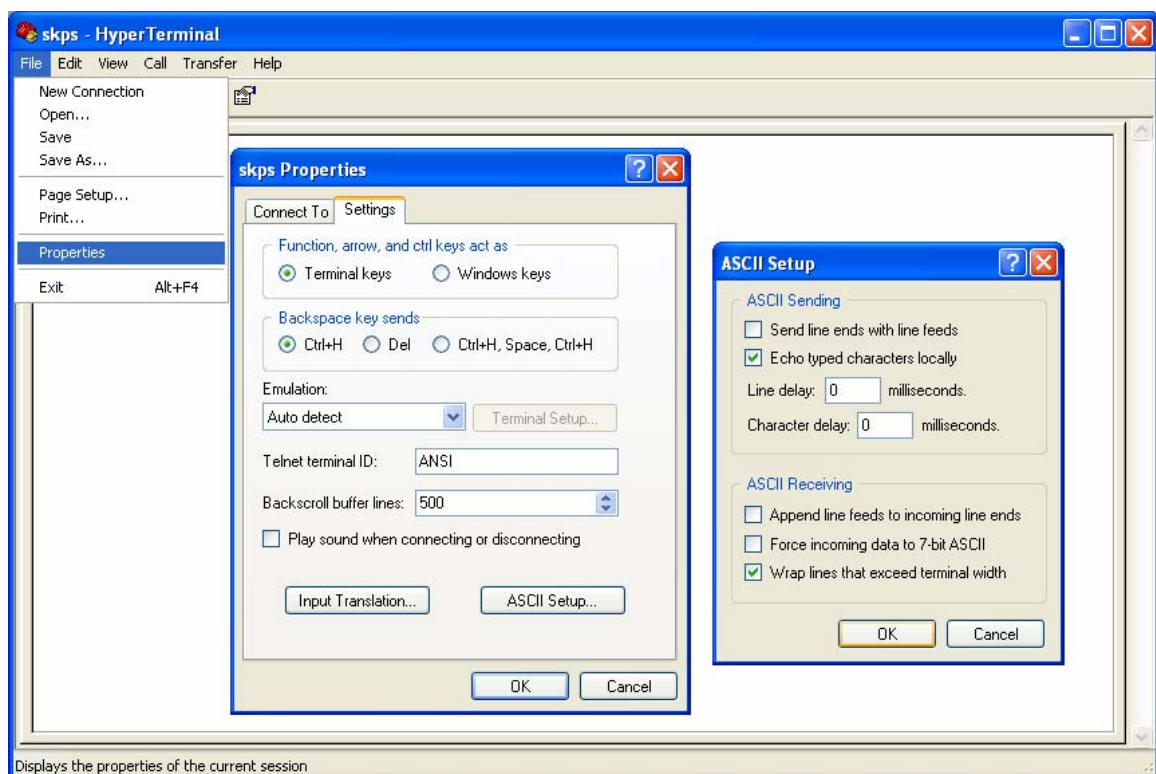
- g. At Device Manager Table, choose Ports (COM & LPT) and you can check your USB Serial Port COM. The Serial Port COM is “COM20” in this case.



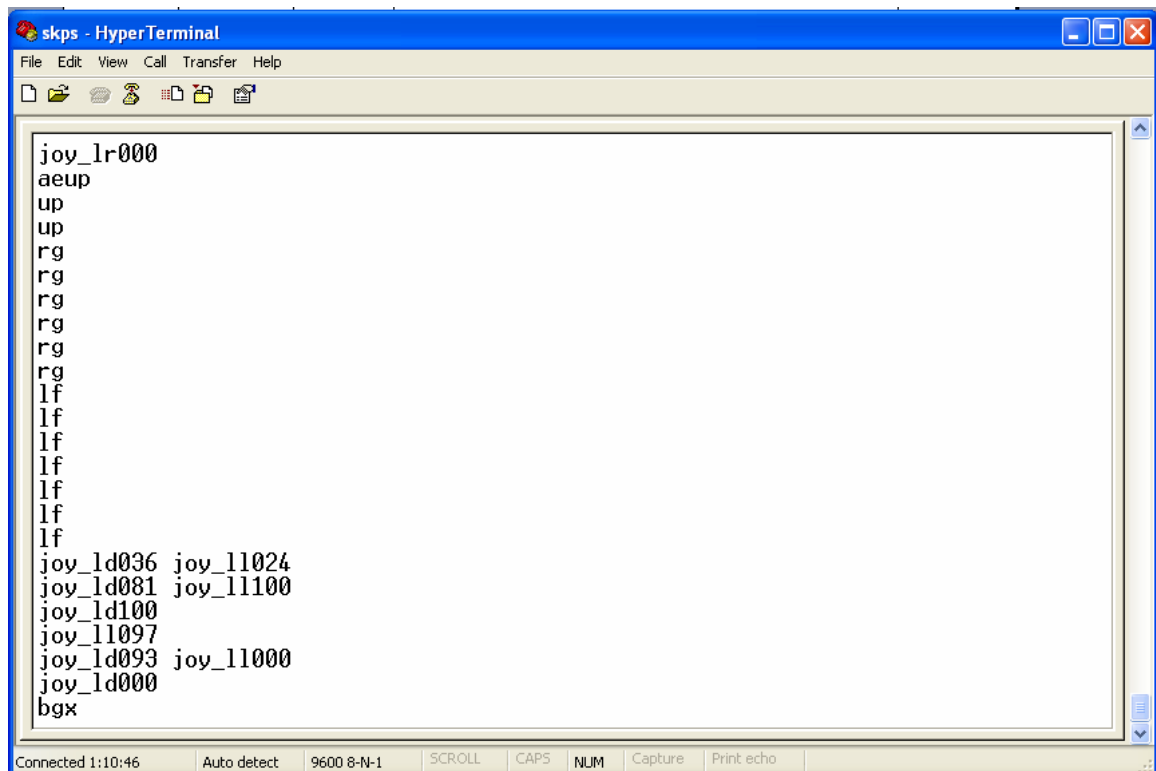
- h. Set the Port Setting as picture below. Bits per second must be same with SKPS Baud Rate and Flow control must be set to none. After finish setting, click Apply and then click OK.



- i. Go to File and select Properties. 'skps' Properties table will show. Choose Setting tab and click ASCII Setup tab. Click on Echo typed characters locally and then click OK.



- j. After all settings are complete, user may now check the functionality of SKPS. Please refer ASCII Mode (PC) protocol in Section 6.2. Type “ae” to enable the return of any key pressed or changes of joystick automatically. Type “ad” to disable this feature.



```
joy_lr000
aeup
up
up
rg
rg
rg
rg
rg
rg
lf
lf
lf
lf
lf
lf
lf
joy_ld036 joy_ll024
joy_ld081 joy_ll100
joy_ld100
joy_ll097
joy_ld093 joy_ll000
joy_ld000
bgx
```

- k. Make sure data is sent in correct protocol by SKPS when button or joystick is being manipulated (press or push). If the data can be obtained, the SKPS is working.

6.3 PS2 Controller

User may choose either wired or wireless PS2 Controller to be connected to SKPS. Figure below shows the method to connect PS2 controller to SKPS board. There are many types of PS controller in the market and the sensitivity for each type also different. User is advised to use original PS2 controller. Cytron Technologies does not guarantee compatibility for all PS2 controllers.



(a) Wireless Connection



(b) Wired Connection

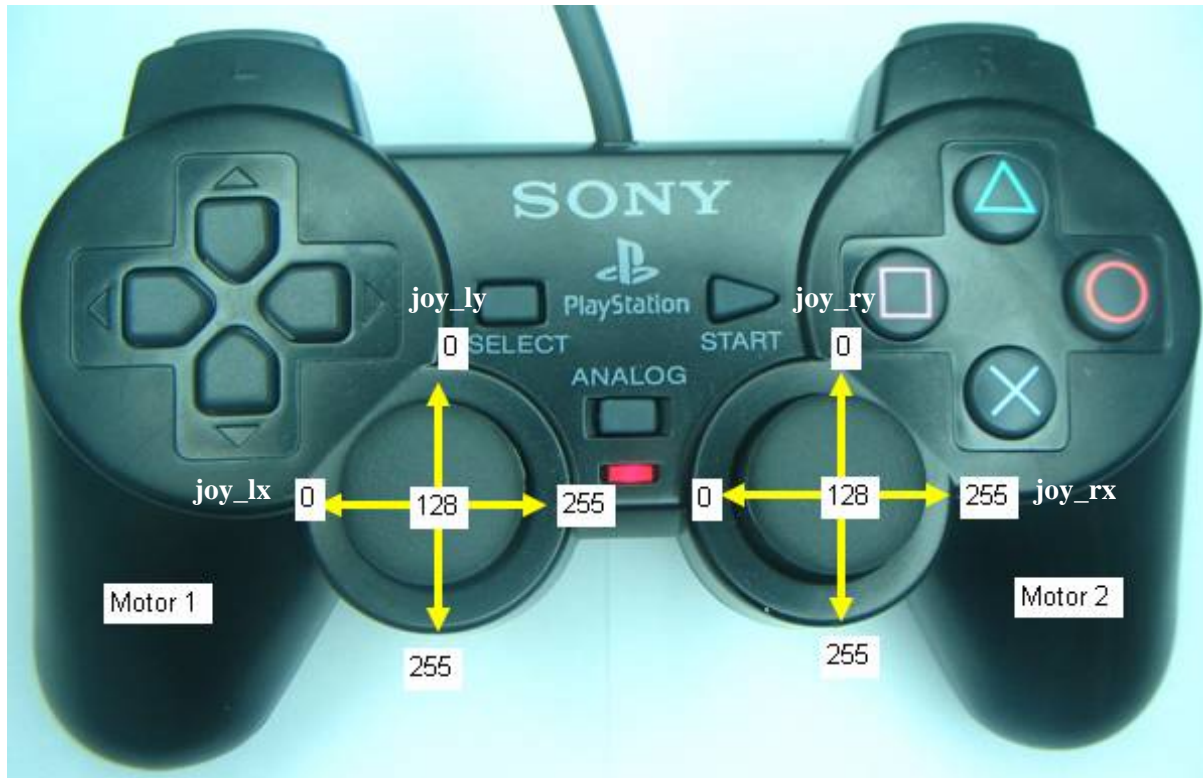
Figure below shows an example of PS2 controller which can be used for SKPS. There are 16 buttons that can be used as input button in PS2 controller.



Analog value of Left Joystick and Right Joystick can be read from SKPS. There are 2 axes that can be read which are Type 1 and Type 2. It will be mentioned briefly after this. User can use either Type 1 or Type 2.

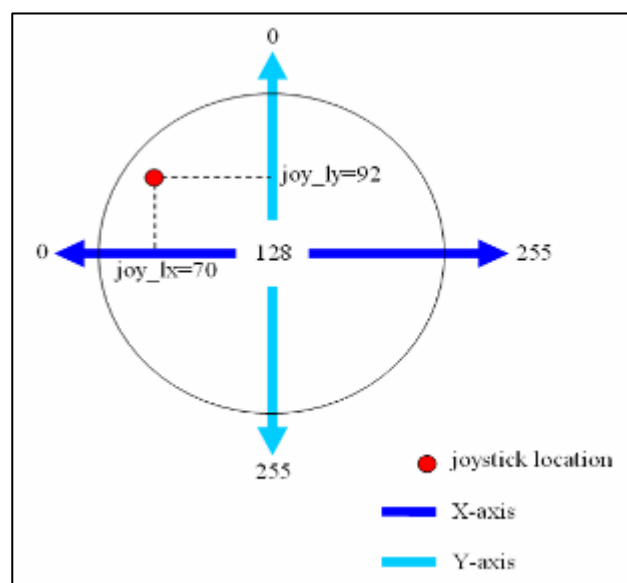
Note: SKPS does not come with PS2 controller, please purchase separately from Cytron Technologies website. It is advised to use PS2 controller from Cytron Technologies because all PS2 controller provided is tested before it is being shipped to customer.

For Type 1, there are 2 variables for each joystick which is axis X and axis Y. For example user can move the joystick up, down for Y axis and left, right for X axis. For Y axis, when users move the joystick up, the value will change from 128 to 0; when user move the joystick down, the value will change from 128 to 255. For X axis, when users move the joystick right, the value will change from 128 to 255; when user move the joystick left, the value will change from 128 to 0. This axis is same for left joystick and right joystick on PS2 controller.

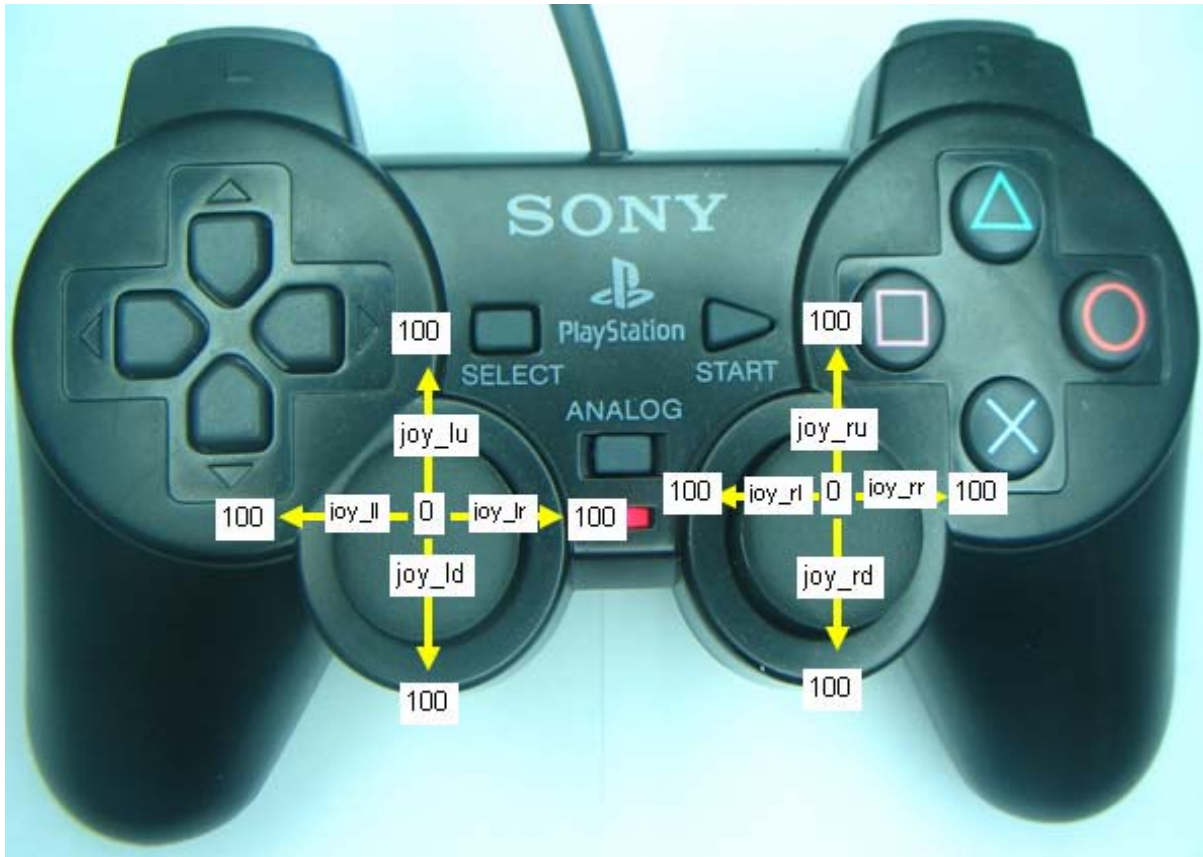


(a) Type 1

Let's take an example on left joystick. Referring to the figure below, the left joystick is moved a bit to the left and a bit to the up direction. So, the value of joy_lx and joy_ly will be changed, as in the figure joy_lx=70 and joy_ly=92.

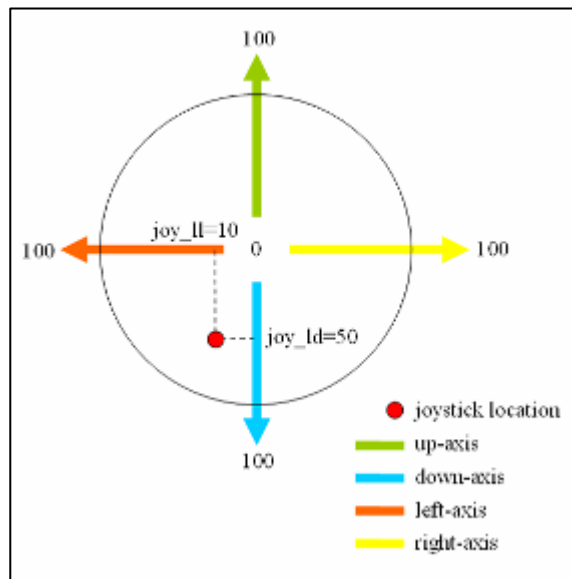


For type 2, they are 4 variables for each joystick. The 4 variables are up, down, left and right. When users move the joystick up, down, left or right the value is at range 0-100. This axis is same for left joystick and right joystick on PS controller.

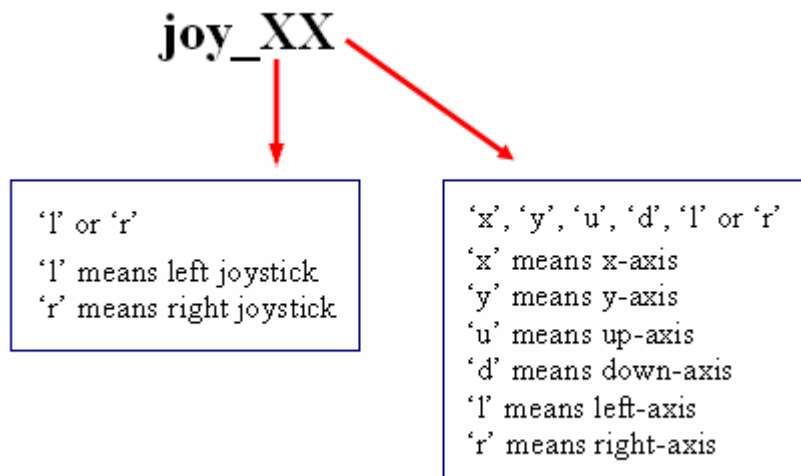


(b) Type 2

For example, when left joystick is being move 10% to left and 50% to down position, the value of joy_ll and joy_ld will change, as in figure below, joy_ll=10 and joy_ld=50. However the value of joy_lu=0 and joy_lr=0.



Please take note that the joy_lx, joy_ly, joy_rx and joy_ry is for type 1 axis; joy_lu, joy_ld, joy_ll, joy_lr, joy_ru, joy_rd, joy_rl and joy_rr is for type 2 axis. This variable is being name according to a standard format.



7. PROTOCOL

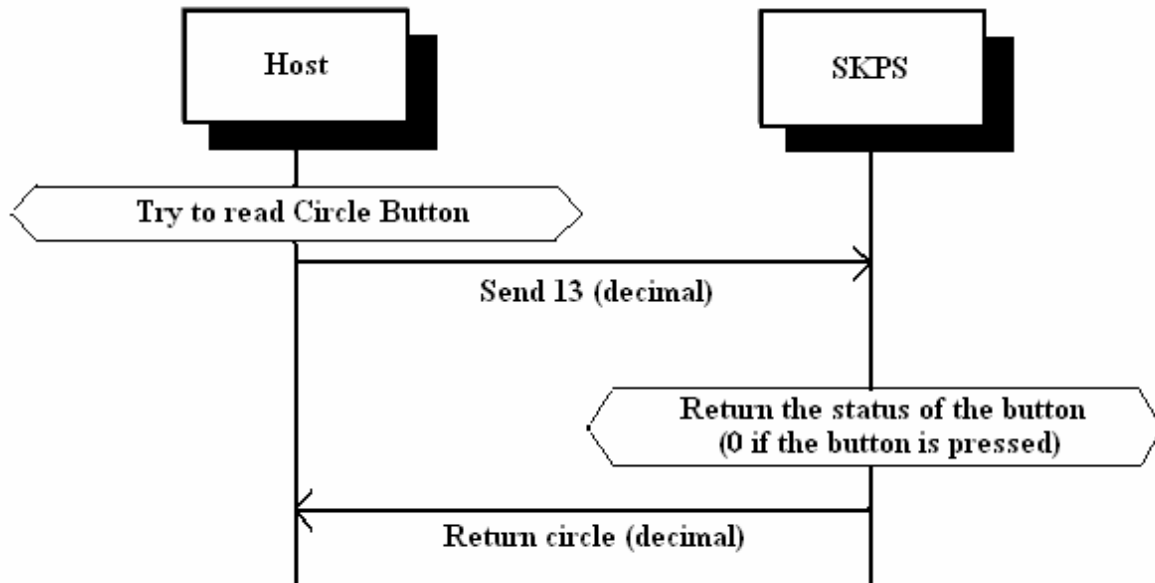
7.1 Non-ASCII Mode (Microcontroller)

7.1.1 Button and PS2 Controller Status

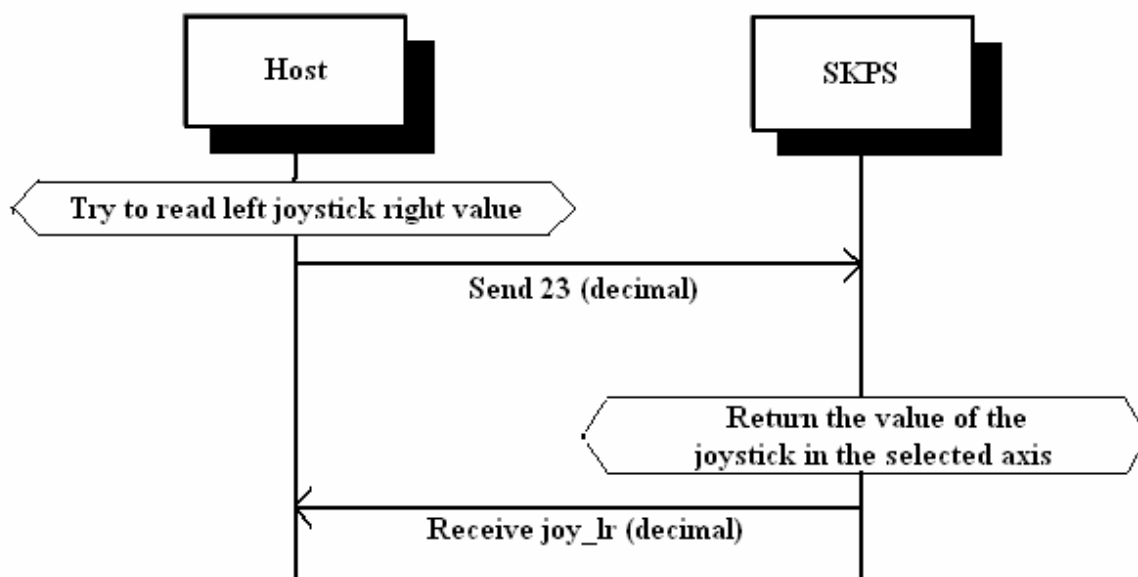
Send (decimal)	Button of Controller	Description
0	select button	<p>SKPS will return the status of corresponding button when the particular decimal value is received 0 if the button is pressed 1 if button is not pressed</p> <p>Example: If microcontroller sent 10 (decimal), SKPS will check L1 button status on PS2 Controller. SKPS will return 0 if L1 button is pressed or 1 if L1 is not pressed.</p>
1	left joystick center button	
2	right joystick center button	
3	start button	
4	up button	
5	right button	
6	down button	
7	left button	
8	L2 button	
9	R2 button	
10	L1 button	
11	R1 button	
12	triangle button	
13	circle button	
14	cross button	
15	square button	
16	left joystick x-axis	<p>SKPS will return the particular value of corresponding joystick in the selected axis</p> <p>Example : If microcontroller sent 20 (decimal), SKPS will read and return the value of left joystick up.</p>
17	left joystick y-axis	
18	right joystick x-axis	
19	right joystick y-axis	
20	left joystick up value	
21	left joystick down value	
22	left joystick left value	
23	left joystick right value	
24	right joystick up value	
25	right joystick down value	
26	right joystick left value	
27	right joystick right value	
28		SKPS will read controller status 1 is return if controller is detected

7.1.2 On board Vibrator Motor Control

send 1st byte (decimal)	send 2nd byte (decimal)	Controller	Description
29	motor1 value	small vibrator placed on right side	motor1 = 1 or 0 (on or off)
30	motor2 value	big vibrator placed on left side	motor2 = 0 -> 255 (adjustable vibration)



(a) Example Communication of Circle Button Operation



(b) Example Communication of Left Joystick Right Value Operation

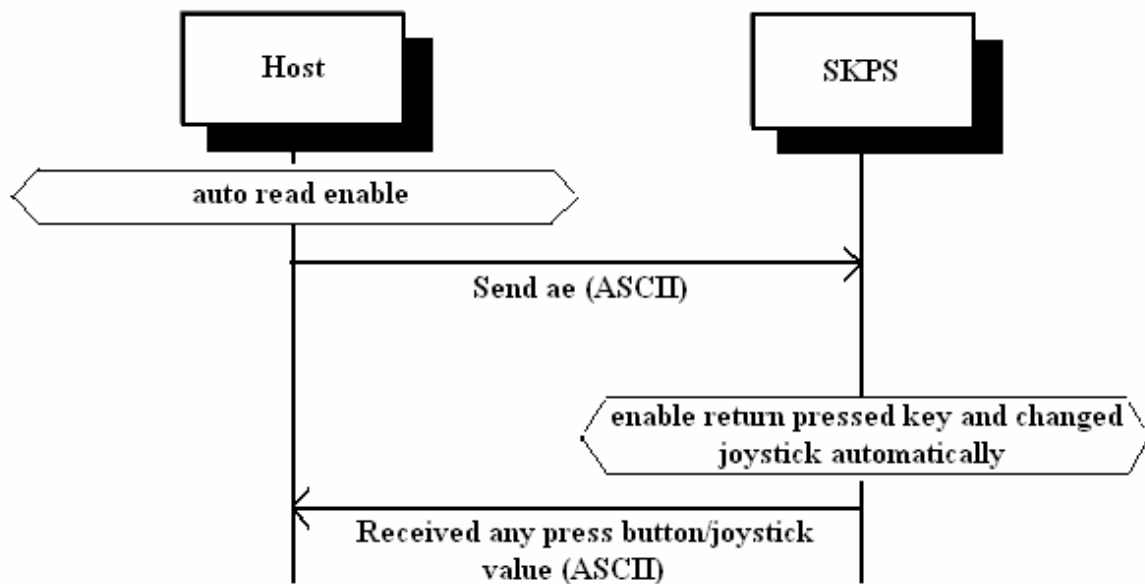
7.2 ASCII Mode (PC)

7.2.1 Button and PS2 Controller Status

Send 2 byte	Button of Controller	Description
up	up button	<p>SKPS will return the status of corresponding button when the particular ASCII value is received '0' if the button is pressed '1' if button is not pressed</p> <p>Example: If microcontroller sent 'r2' (ASCII), SKPS will check R2 button status on PS2 Controller. SKPS will return '0' if R2 button is pressed or '1' if R2 is not pressed. ["Enter" after every result]</p>
dw	down button	
lf	left button	
rg	right button	
tr	triangle button	
cr	circle button	
sq	cross button	
ci	square button	
st	start button	
se	select button	
jl	left joystick center button	
jr	right joystick center button	
l1	L1 button	
l2	L2 button	
r1	R1 button	
r2	R2 button	
lx	left joystick x-axis	<p>SKPS will return the particular value of corresponding joystick in the selected axis (3 byte, in ASCII). Example: '235' or '029'. ["Enter" after every result]</p>
ly	left joystick y-axis	
rx	right joystick x-axis	
ry	right joystick y-axis	
lu	left joystick up value	<p>SKPS will return the particular value of corresponding joystick in the selected axis (3 byte, in ASCII). Example: '039' or '027'. ["Enter" after every result]</p>
ld	left joystick down value	
ll	left joystick left value	
lr	left joystick right value	
ru	right joystick up value	
rd	right joystick down value	
rl	right joystick left value	
rr	right joystick right value	
cs		SKPS will read controller status '1' is return if controller is detected
unknown		SKPS will return 'x' if unknown data is sent.

7.2.2 Read Key Functions

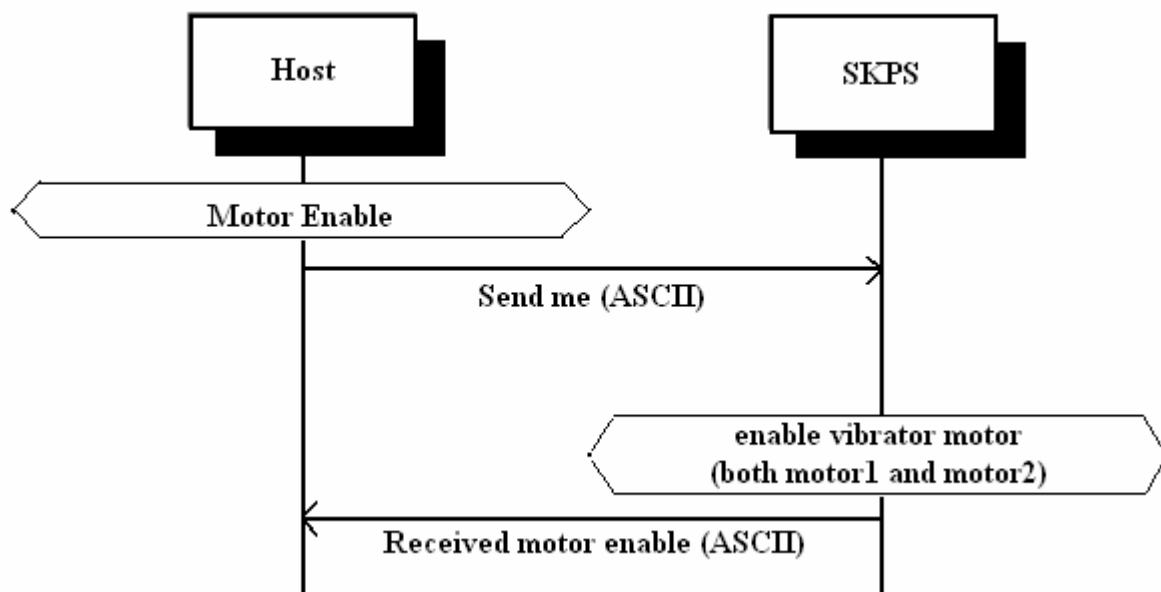
Send 2 byte	Received	Controller	Description
rk	all key status	any press button/joystick value	SKPS will return pressed key and changed joystick value once. [“Enter” after result]
ae	auto read enable	any press button/joystick value	SKPS will enable return pressed key and changed joystick automatically. [“Enter” after result]
ad	auto read disable	any press button/joystick value	SKPS will disable return pressed key and changed joystick automatically. [“Enter” after result]



Example communication of Auto Read Enable Operation

7.2.3 On board Vibrator Motor Control

Send 2 byte	Received	Description
me	motor enable	SKPS will enable vibrator motor (both motor1 and motor2)
md	motor disable	SKPS will disable vibrator motor



Example Communication of Motor Enable Operation.

8. GETTING STARTED

For this section, SKPS will be interfaced with PR23. Please refer PR23, DIY project from Cytron website for details example of interfacing to SKPS. PR23 shows the method of using SKXBee, since SKPS UART pin is designed to be compatible to SKXBee, user may replace SKXBee with SKPS on PR23. Please refer to <http://www.cytron.com.my/PR23.asp> for the details of PR23. For method of writing most simple program to use with SKPS, please refer steps below and for sample full version source code, user may get it from Cytron Website (same page as SKPS).

- a. Set the configuration for UART. Make sure the baud rate is correct.

```
//setup USART
SPBRG = 0x81;           //set baud rate to 9600 for 20Mhz
BRGH = 1;               //baud rate high speed option
TXEN = 1;               //enable transmission
TX9 = 0;
CREN = 1;               //enable reception
SPEN = 1;               //enable serial port
RX9 = 0;
RCIE = 0;               //disable interrupt on eachdata received
```

- b. Definition of every button and joystick. For following example, “p_” is added in front of PS2 controller label for easy understandable and avoid crash with other variable or instructions.

```
//skps protocol
#define p_select        0
#define p_joy1          1
#define p_joyr          2
#define p_start         3
#define p_up            4
#define p_right         5
#define p_down          6
#define p_left          7
#define p_l2            8
#define p_r2            9
#define p_l1            10
#define p_rl            11
#define p_triangle      12
#define p_circle        13
#define p_cross         14
#define p_square        15
#define p_joy_lx         16
#define p_joy_ly         17
#define p_joy_rx         18
#define p_joy_ry         19
#define p_joy_lu         20
#define p_joy_ld         21
#define p_joy_ll         22
#define p_joy_lr         23
#define p_joy_ru         24
#define p_joy_rd         25
#define p_joy_rl         26
#define p_joy_rr         27

#define p_con_status     28
#define p_motor1         29
#define p_motor2         30
```

- c. Function to read SKPS and control the joystick. Below are examples of UART function and SKPS function. Comments explain each command. UART function controls process of sending and receiving data via UART while SKPS function is to read information on PS2 controller including function to read button and joystick and control the vibrator motor.

```
// uart function
//=====
void uart_send(unsigned char data) //function to send out a byte via uart
{
    while(TXIF==0); //wait for previous data to finish send out
    TXREG=data; //send new data
}

unsigned char uart_rec(void) //function to wait for a byte receive from uart
{
    unsigned char temp;
    while(RCIF==0); //wait for data to received
    temp=RCREG;
    return temp; //return the received data
}

// skps function
//=====
unsigned char skps(unsigned char data) //function to read button and joystick
//information on ps controller
{
    uart_send(data);
    return uart_rec();
}

unsigned char skps_vibrate(unsigned char motor, unsigned char value)
//function to control the vibrator motor
//on ps controller
{
    uart_send(motor);
    uart_send(value);
}
```

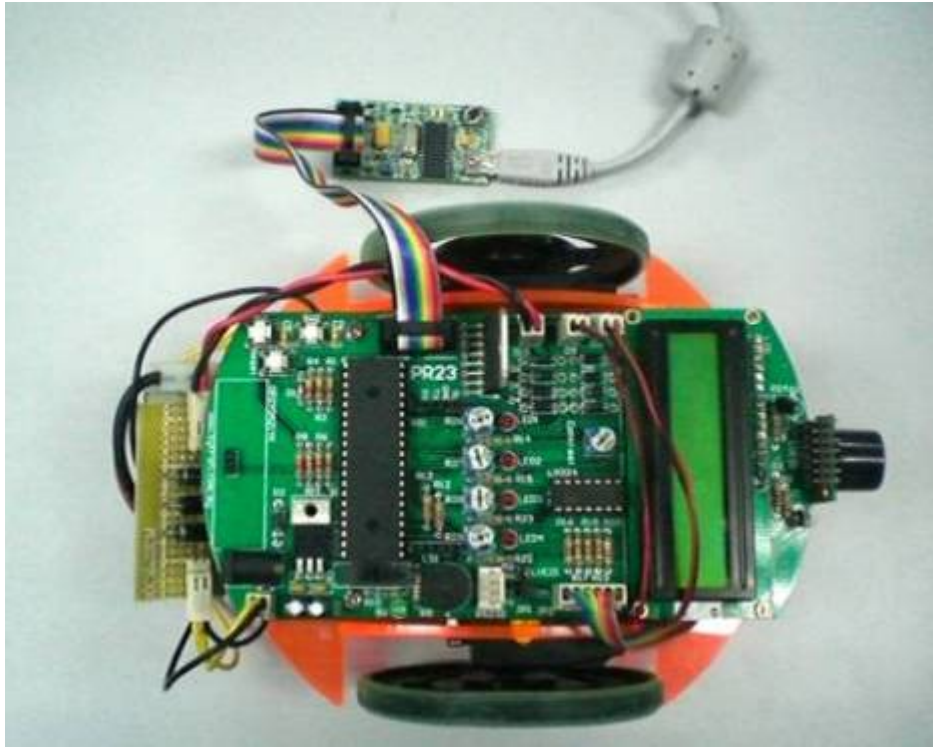
- d. Below are sample on how to use those function that mentioned above. For first command, when L1 is pressed, buzzer will beep. Second command is to test if there is any obstacle in front. For PR23 we use ultrasonic sensors to detect the obstacles. If obstacle is detected, motor in PS2 controller will vibrate.

```
//test button for horn
if(skps(p_l1)==0)buzzer=1; //if button L1 pressed, beep the buzzer
else buzzer=0;

//test if there is any obstacle in front
if(read_adc(CHANNEL0)<20) //if ultrasonic sensors detect obstacle in front,
{ //vibrate the ps controller
    skps_vibrate(p_motor2,255);
}
else skps_vibrate(p_motor2,0);
```

NOTE: User may download this sample source code from product page of SKPS.

- e. After finish writing the program, user may compile it and load the source code. Please refer to PR23 on the method to download the code.



- f. Upon loading the program, PR23 is now ready to operate with SKPS. User may test it using PS2 controller, try each button and function. Have fun!



9. WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damage caused by mis-use is not covered under warranty.
- Warranty does not cover freight cost for both ways.

Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: +607-521 3178

Fax: +607-521 1861

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my